Project Work and Internship

**REPORT**

Accident Causes classification

horizontal line

# Abstract

In this study, text mining techniques along with deep neural networks are used for text classification. In detail, GloVe word embedding with 300 dimensions are used for accident reports classification. Dataset from Occupational Safety and Health Administration is used for evaluating performance of the proposed model. Besides, 5 baseline models namely K-Nearest Neighbours, Support Vector Machines, Logistic Regression, Naive Bayes and Decision Tree are trained with tf-idf vectorized sentences to compare with the proposed approach. Experiment results show that the proposed approach achieves the highest average weighted F1 score among all the models considered in this study. The result also proves that using GloVe embeddings captured semantic relation between words and extracting of noun and verb phrases helped in boosting the performance.

## Data

The proposed model has wide varieties of datasets ranging from 11 different classes for Accident Causes Classification. There were majorly two different categories which includes: Labelled Dataset Category and the Unlabelled Datasets which includes 1,000 and 16,323 entries respectively. For Accident Causes Classification, Labelled Datasets have been taken into account and several text-preprocessing algorithms along with model formation have been applied. After further analysis, it was noted that the Labelled Dataset Category contains slightly biased categories which is making it imbalanced. Hence, manual addition of Datasets in these categories are still undergoing. The team has decided to increase data in the category of “Exposure to extreme Temperatures”, ”Exposure to chemical Substances”, ”Struck by falling object”, ”Struck by moving objects” and ”Traffic”. Some resources have been found and for rest the work is still undergoing.

**Preprocessing**

Preprocessing the data is one of the very crucial task in generating accuracy and predictions of the model. Keeping this goal in mind, the team has applied wide varieties of text preprocessing algorithms.

* **String Lowercase**: All the uppercase textual data have been brought to a similar lowercase context.
* **Removal of Punctuations and Numbers**: All the unnecessary punctuation and numerical records have been eliminated from the text corpus.
* **Tokenization:** The set of textual data after the above steps have now been converted into a set of tokens . For achieving this,the team has utilised Keras and nltk libraries.
* **Stop Words Removal:** A domain specific stopword removal scheme has been applied onto a set of text corpus where it removed noise from the data. The domain specific stopwords include all the english language based stopwords along with months of the year.
* **POS tagging:** Followed by this, a POS tagging scheme has been applied onto the text corpus and words were tagged along with their Parts of Speech Data. This was done to ensure , extract and chunk out noun and verb based phrases. These noun and verb based phrases proved to be more significant in interpreting textual semantics.
* **Lemmatization:** Followed by this, Lemmatization has been applied on the pre-processed text corpus to bring the word to it’s base/root form.
* **TF-IDF and Label Encoder:** For encoding, TF-IDF scheme has been adopted to convert the preprocessed textual data into numbers. For this, the n\_gram range of 1,2 has been adopted which groups the unigrams and bigrams both. This has been applied on the Independent Variable i.e “Summary2” column of Tagged Datasets. For the output class of the “Dependent Variable Category” i.e the “Tagged2” Column, Label Encoder have been used.
* **Text to Sequences:** Each word in the corpus assigned an id and each preprocessed report in the data is converted into sequences of word ids and padded with zeros for the same length of 200 numbers for each report.

**Glove Word Embeddings**

GloVe embeddings are downloaded from the internet having different dimensions like 50,100,200,300 length for each word. Embeddings are visualized using Principal Component Analysis. Embedding matrix is created for words in the corpus of text data.

**Extraction of Noun and Verb Phrases**

Noun and verb phrases are extracted from the title column using two approaches. First one is using textacy library and the second approach is by creating my own grammar. For noun phrases the second approach works better whereas for verb phrases the first approach works better. Created dictionary for frequency of nouns present in text corpus. Most frequent nouns are ladder, scaffold, fire, roof, tree, backhoe, etc. Finally every noun and verb phrase are converted into a sequence of integers and padded for the same length.

**Model**

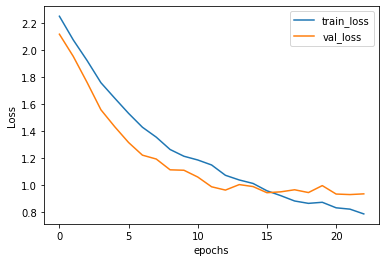
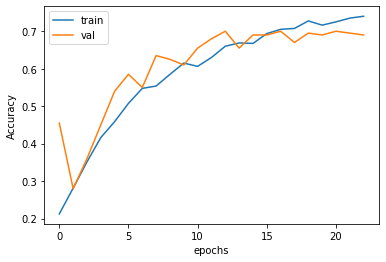
A hybrid structured neural network is designed for classification of text reports. First layer is the embedding layer using the embedding matrix from GloVe embeddings and preprocessed sequences of text reports. Second layer is a Dropout layer. Third layer is a Batch Normalization layer. Then a Convolution layer with relu activation followed by a MaxPool layer. Then an LSTM layer with tanh activation with dropout. Finally an Attention layer is added followed by a Dense layer with softmax activation to give output. Model is trained on part of the labelled dataset using Adam optimizer with exponential learning rate decay.

The proposed preprocessed texts have also been passed to the traditional baseline models namely SVM, KNN, Logistic Regression, Decision Tree and Naive Bayes.

**Results**

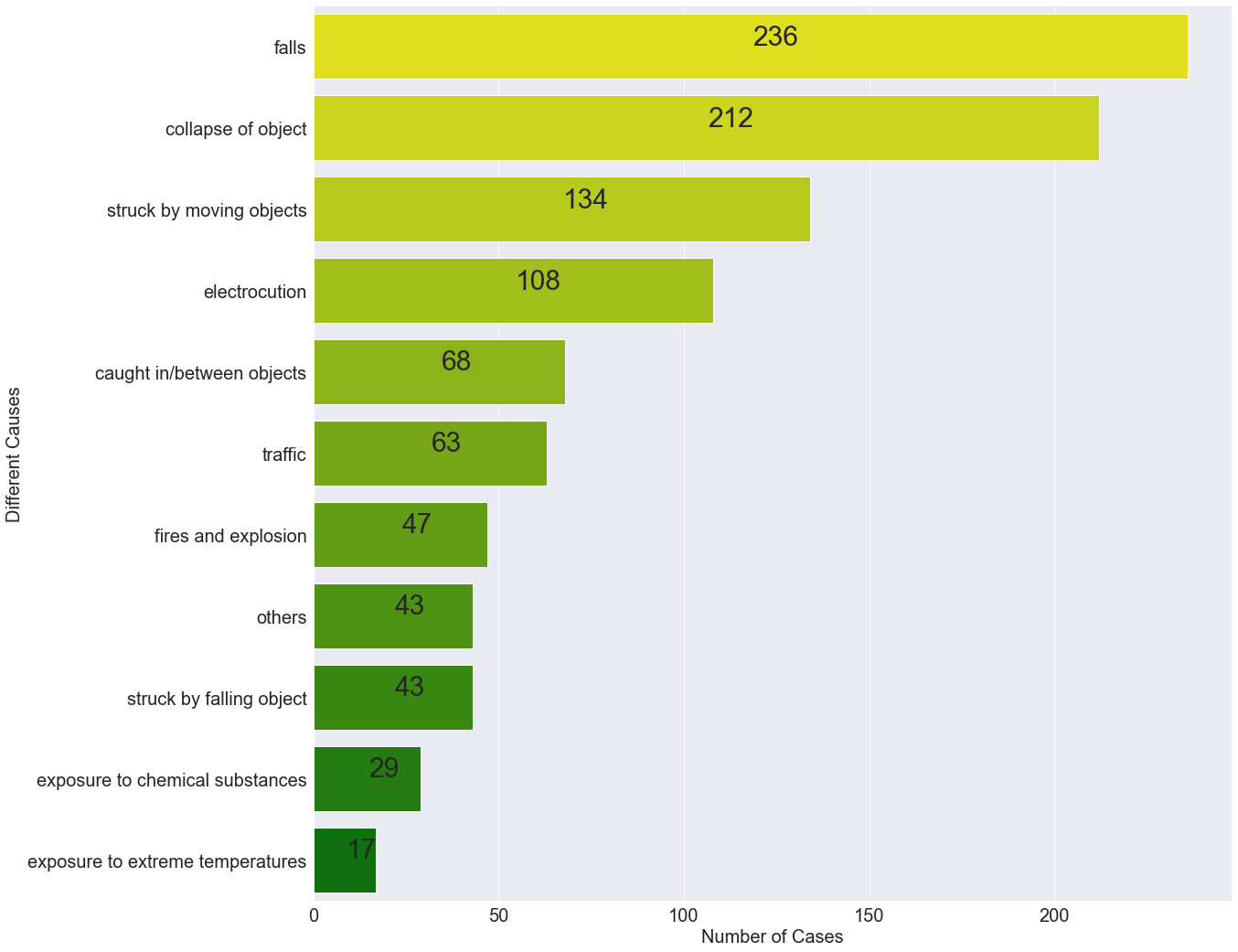
The proposed approach achieved an average weighted F1 score of **0.813** which is highest among all the models used in this study. Apart from these , in the traditional baseline models **SVM** was used. It gave an average weighted F1 score of **0.556** with Linear Kernel and ngrams included are both unigrams and bigrams. In case of **Logistic Regression** it gave the result of **0.402** as average weighted F1 score whereas in **KNN** it was **0.480** with Nearest Neighbour as 5. In the case of **Naive Bayes** , it’s performing worse among all. It gave an average weighted F1 score as **0.390** and in case of **Decision Tree** it was **0.498**.

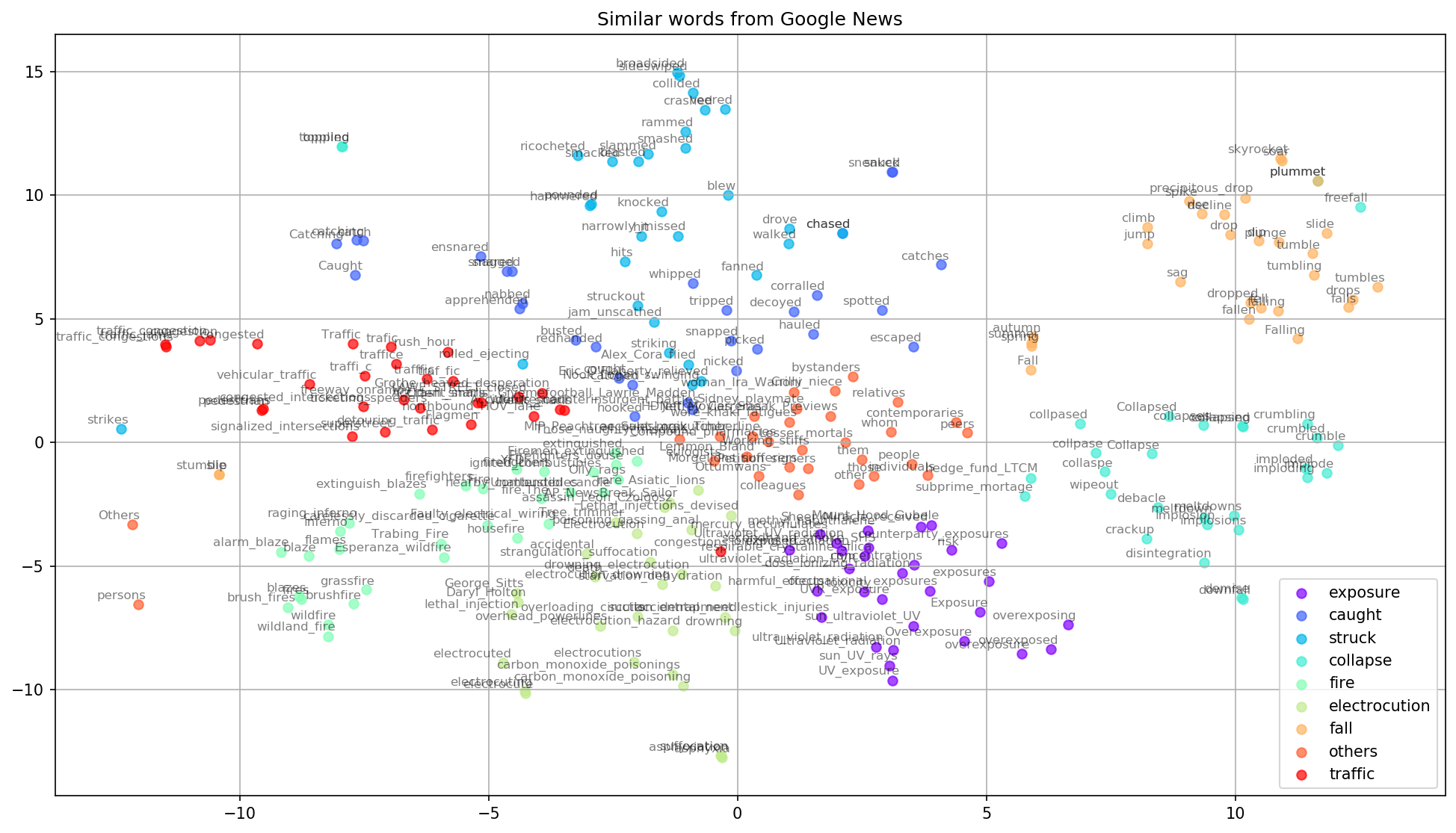
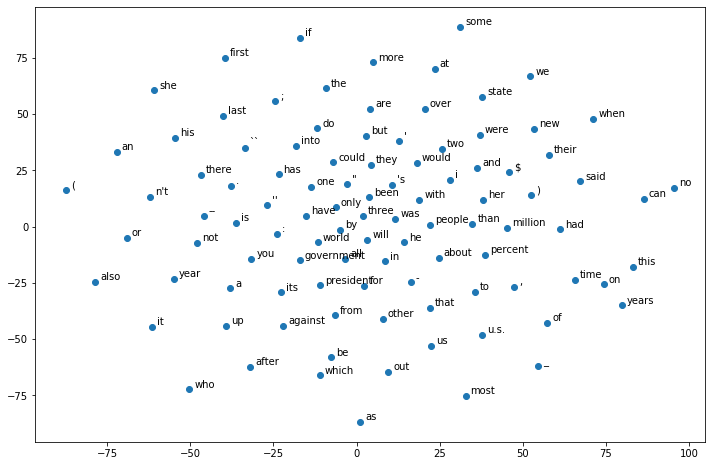
**Loss and accuracy plots for the proposed approach are:**



**Future Works(in next 15 days)**

* The next days are targeted at gaining and collecting vivid datasets in the above mentioned categories and making it a balanced version.
* Using bigrams and trigrams of GloVe for training the proposed model.
* Working on the transformer and multi head attention model to add into the proposed approach.

**Distribution of labels for labelled data**

1. **GloVe Word Embeddings, 2) Most similar words to labels in data **

**Word Cloud for Unlabelled data Noun Phrases frequency**

**MODEL SUMMARY (Hybrid Neural Network with Attention)**

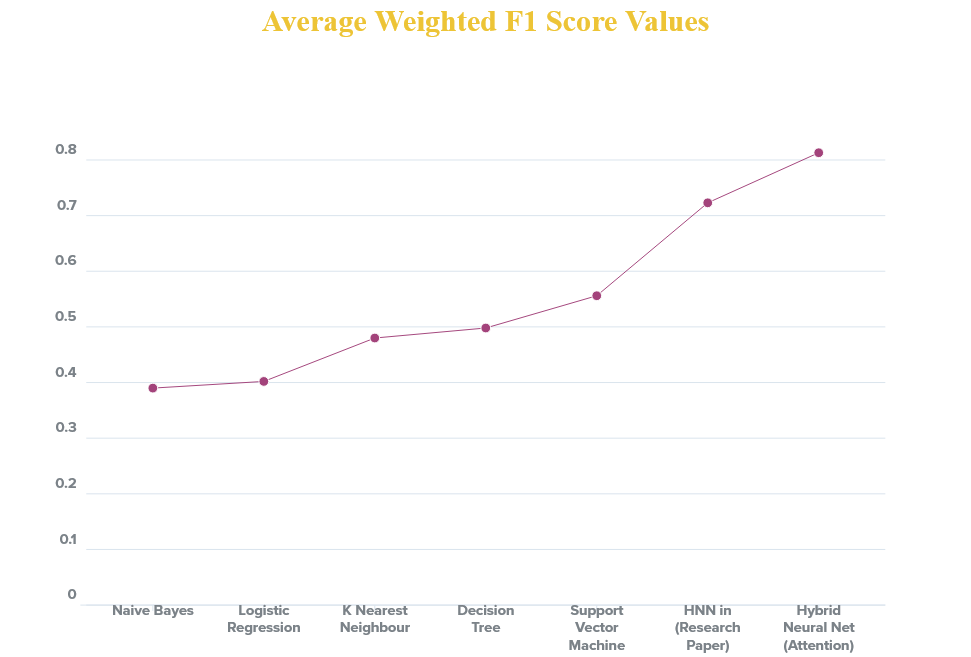
|  |  |  |
| --- | --- | --- |
| **Layer(type)** | **Output Shape** | **Param #** |
| Embedding | (None, 212, 300) | 6081000 |
| Dropout | (None , 212, 300) | 0 |
| Batch Normalization | (None, 212, 300) | 1200 |
| Conv1D | (None, 208, 30) | 45030 |
| MaxPooling1D | (None, 104, 30) | 0 |
| LSTM | (None, 104, 30) | 7320 |
| Attention | (None, 30) | 134 |
| Dense | (None, 11) | 341 |

Total Params: 6,135,025

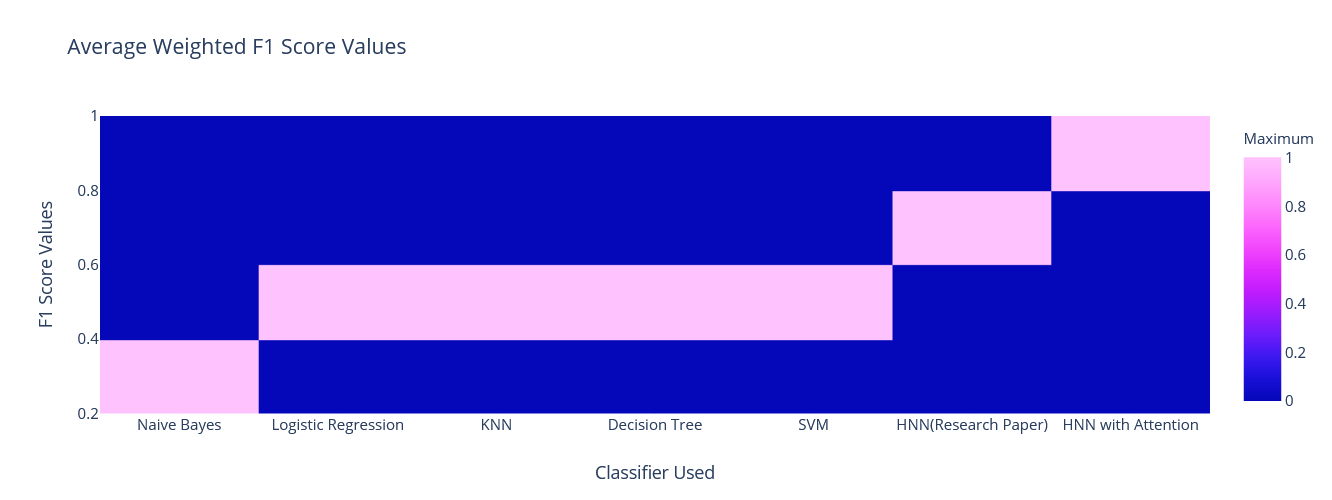
Trainable Params: 53,425

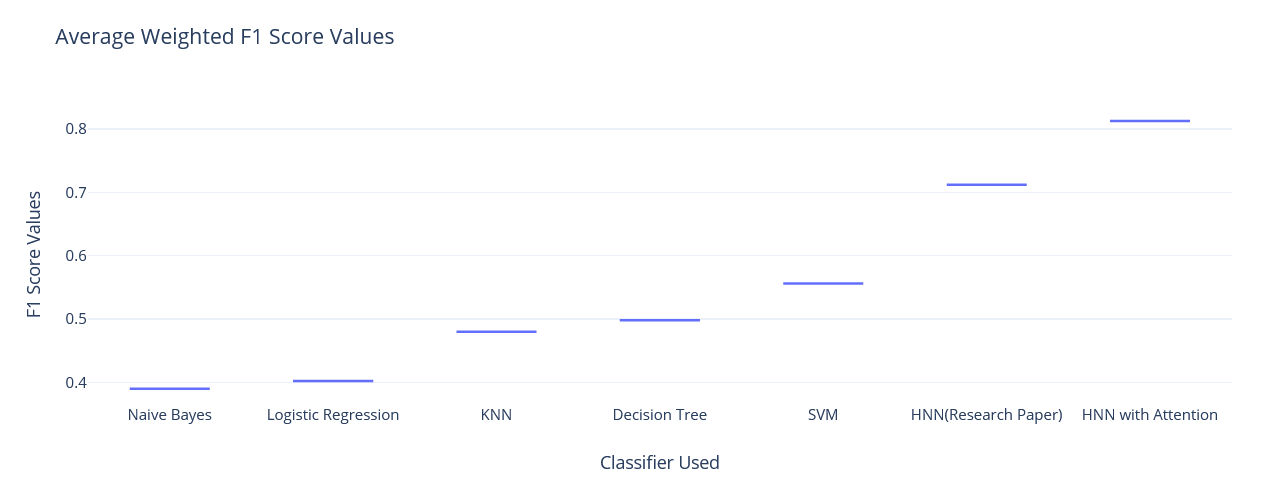
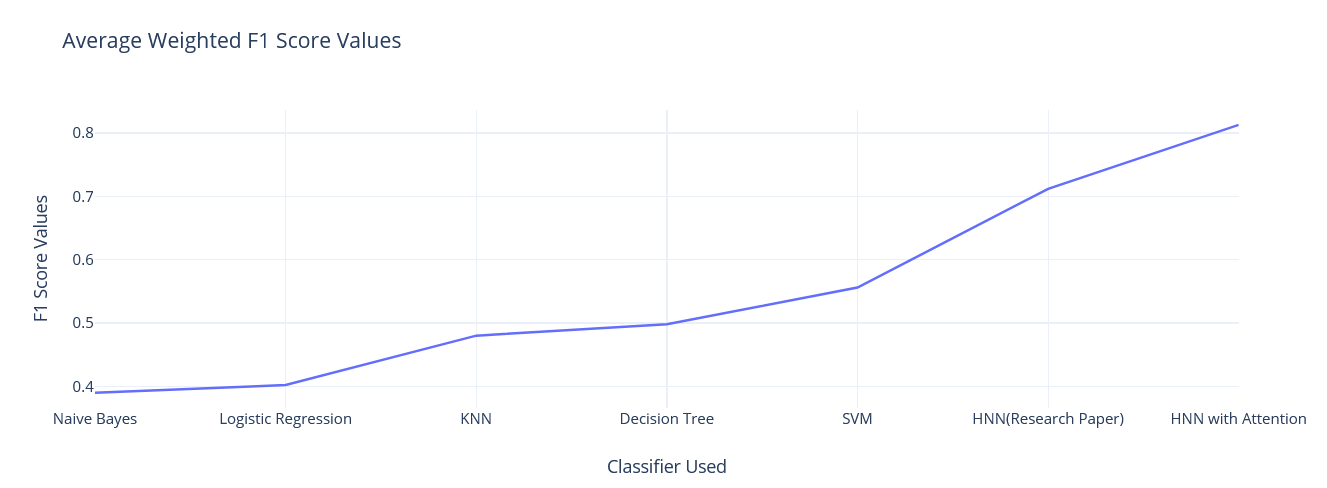
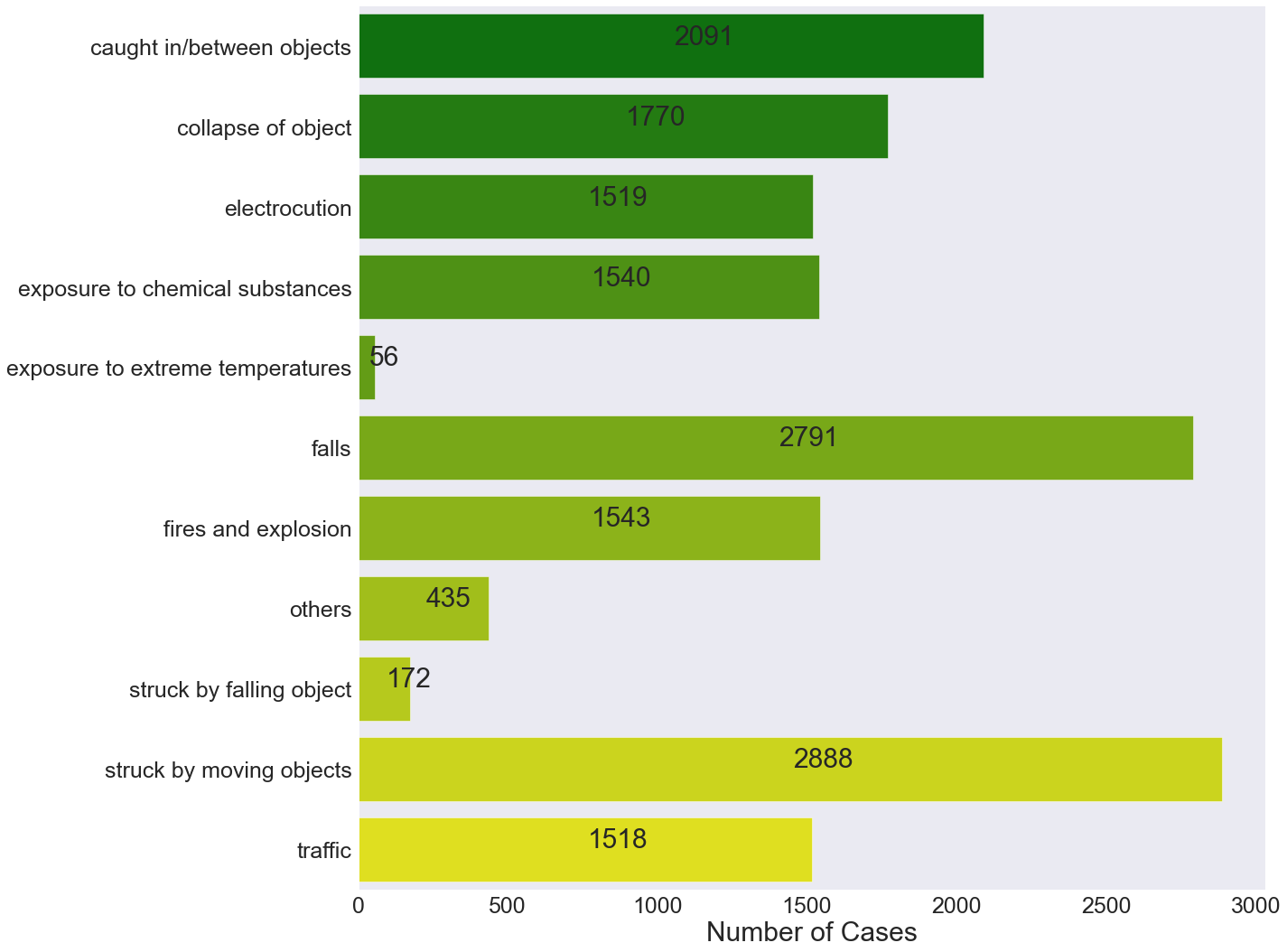
Non Trainable Params: 6,081,600

**Average Weighted F1 Score Values**

****

**2D Histogram Showing Average Weighted F1 Score Values**

****

**Predictions on Unlabelled data**